BÁO CÁO THỰC HÀNH LAB 3

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

**Basic Object-Oriented Techniques**

Table of Contents

[1. Branch your repository 2](#_Toc120372917)

[2. Working with method overloading 3](#_Toc120372918)

[2.1 Overloading by differing types of parameter 3](#_Toc120372919)

[2.2. Overloading by differing the number of parameters 4](#_Toc120372920)

[3. Passing parameter 4](#_Toc120372921)

[4. Use debug run 7](#_Toc120372922)

[4.1 Setting, deleting & deactivate breakpoints: 8](#_Toc120372923)

[4.2 Run in Debug mode 8](#_Toc120372924)

[4.3 Investigate value of variables: 9](#_Toc120372925)

[4.4 Change value of variables 9](#_Toc120372926)

[5. Classifier Member and Instance Member 11](#_Toc120372927)

[6. Create accessors and mutators for the class DigitalVideoDisc 12](#_Toc120372928)

[Open the Cart class 12](#_Toc120372929)

[7. Implement the Store class 17](#_Toc120372930)

[8. Re-organize your projects 19](#_Toc120372931)

[9. String, StringBuilder and StringBuffer 20](#_Toc120372932)

# 1. Branch your repository

* Create a branch **refactor/apply-release-flow** for refactoring the repository following the Release Flow
* Create a branch **topic/method-overloading** for the exercise on method overloading
* Create a branch **topic/passing-parameter** for the exercise where you investigate on Java’s parameter passing
* Create a branch **topic/class-members** for the exercise where you practice with classifier member and instance member
* Create a branch **feature/print-cart** for the implementation of the print items in cart feature
* Create a branch **feature/search-cart** for the implementation of the search items in cart feature
* Create a branch **topic/store** for the implementation of the class Store
* Create a branch **refactor/packages** for refactoring the projects in your repository using packages
* Create a branch **topic/memory-management-string** for the String, StringBufer & StringBuilder exercise

# 2. Working with method overloading

## 2.1 Overloading by differing types of parameter

+ You will create a new method that has the same name but with different type of parameter.

**addDigitalVideoDisc(DigitalVideoDisc [] dvdList)**



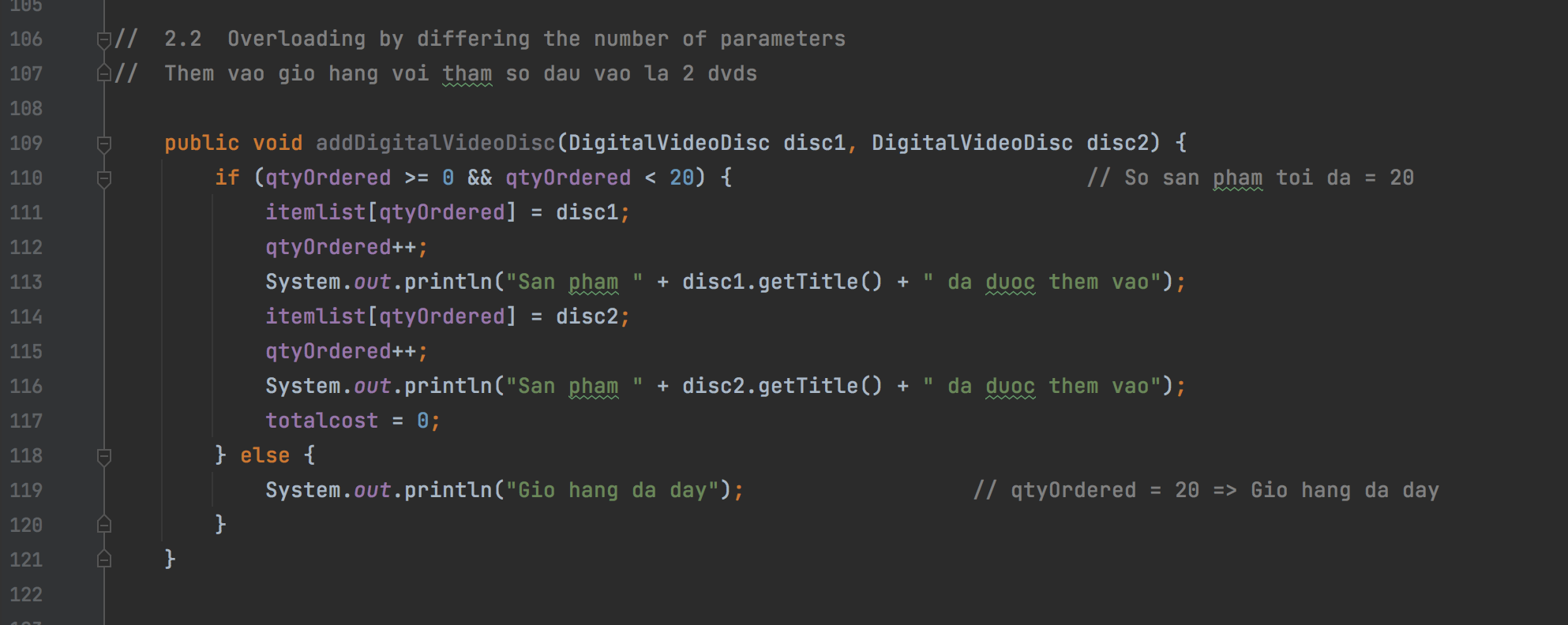
+ Try to add a method addDigitalVideoDisc which allows to pass an arbitrary number of arguments for dvd. Compare to an array parameter



## 2.2. Overloading by differing the number of parameters

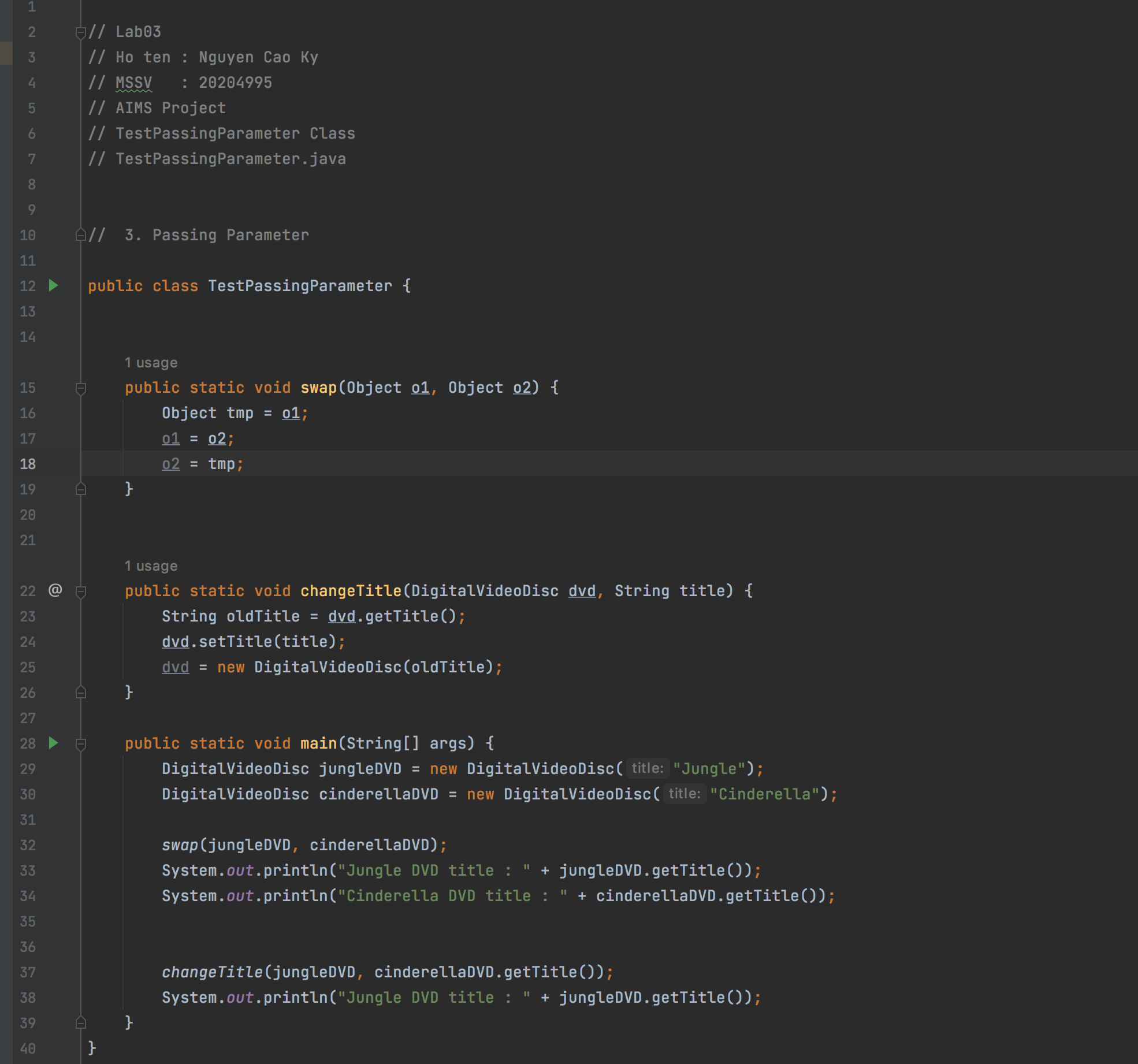
**+** The signature of this method has two parameters as following:

**addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2)**

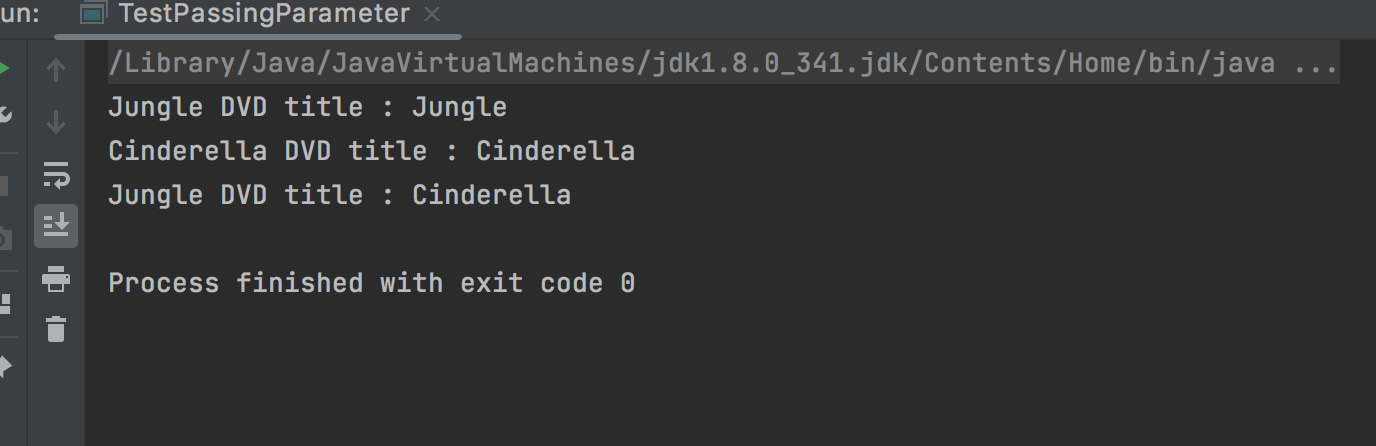


# 3. Passing parameter

TestPassingParameter Class :



Kết quả :



● After the call of swap(jungleDVD, cinderellaDVD) why does the title of these two objects still remain?

● After the call of changeTitle(jungleDVD, cinderellaDVD.getTitle()) why is the title of the JungleDVD changed?

*Giải thích*

**Java chỉ hỗ trợ chuyền tham số kiểu Tham trị**

**+** Đối với phương thức swap () {

Java chỉ sao chép giá trị khi tham số được truyền vào trong một phương thức.  o1 và o2  bên trong swap() không phải o1, o2  ở ngoài, nó chỉ là một biến khác được sao chép ra, cũng mang tên o1 , o2 , cũng mang giá trị giống như o1 , o2 ở ngoài truyền vào, nhưng không còn là o1 , o2 ở ngoài nữa. Do đó dù bên trong swap() chúng ta có thay đổi o1 , o2 như thế nào thì cũng không ảnh hưởng với o1, o2 bên ngoài.

+ Đối với phương thức changeTitle () {

Thực chất vẫn đúng là truyền kiểu Tham trị. Có nghĩa là giá trị của biến dvd được sao chép qua. Nhưng đối tượng dvd ở đây có các thuộc tính như title , category , … Giá trị của một đối tượng lúc bây giờ chính là địa chỉ đến các thành phần của đối tượng đó. Do đó việc sao chép giá trị lúc này cũng như sao chép địa chỉ đến các thành phần title , category ,…

Khi câu lệnh dvd.setTitle(title) được thực hiện , dvd là một đối tượng mới nhưng giá trị của dvd là các địa chỉ đến các thuộc tính của dvd bên ngoài ( cụ thể ở trường hợp này là title ) . Do đó dvd bên ngoài đã được thay đổi tên.

Còn ở câu lệnh dvd = new …

Việc sao chép giá trị vẫn diễn ra , nhưng nó được khởi tạo thành 1 đối tượng mới thông qua toán tử new, thế nên giá trị mới cũng thay đổi theo.

**Please write a swap() method that can correctly swap the two objects.**

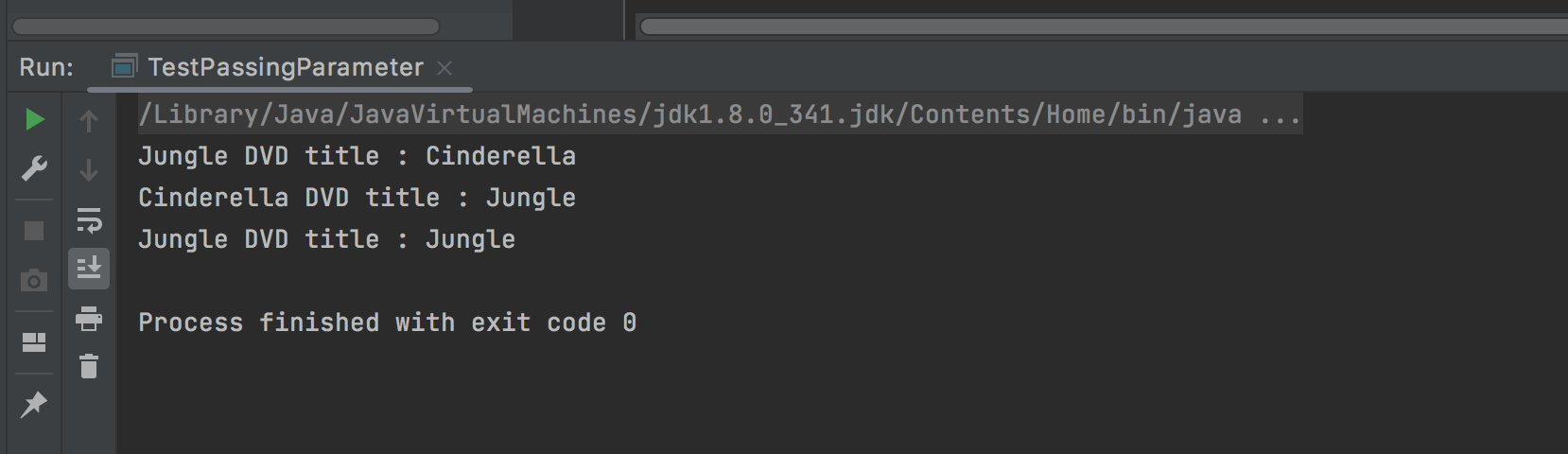
*SwaptwoObj method in DigitalVideoDisc Class :*



Swap method in TestPassingParameter Class:

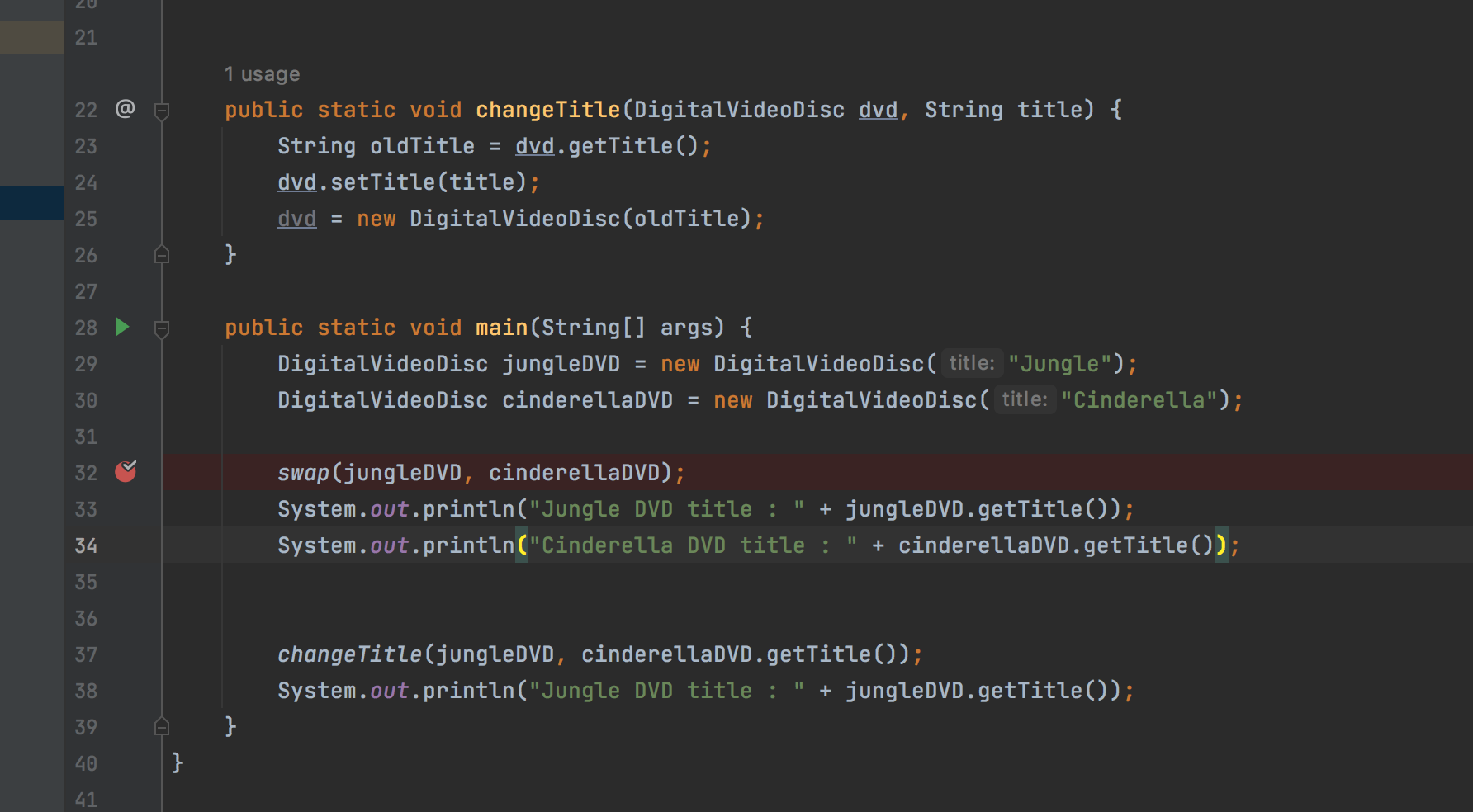


Kết quả:

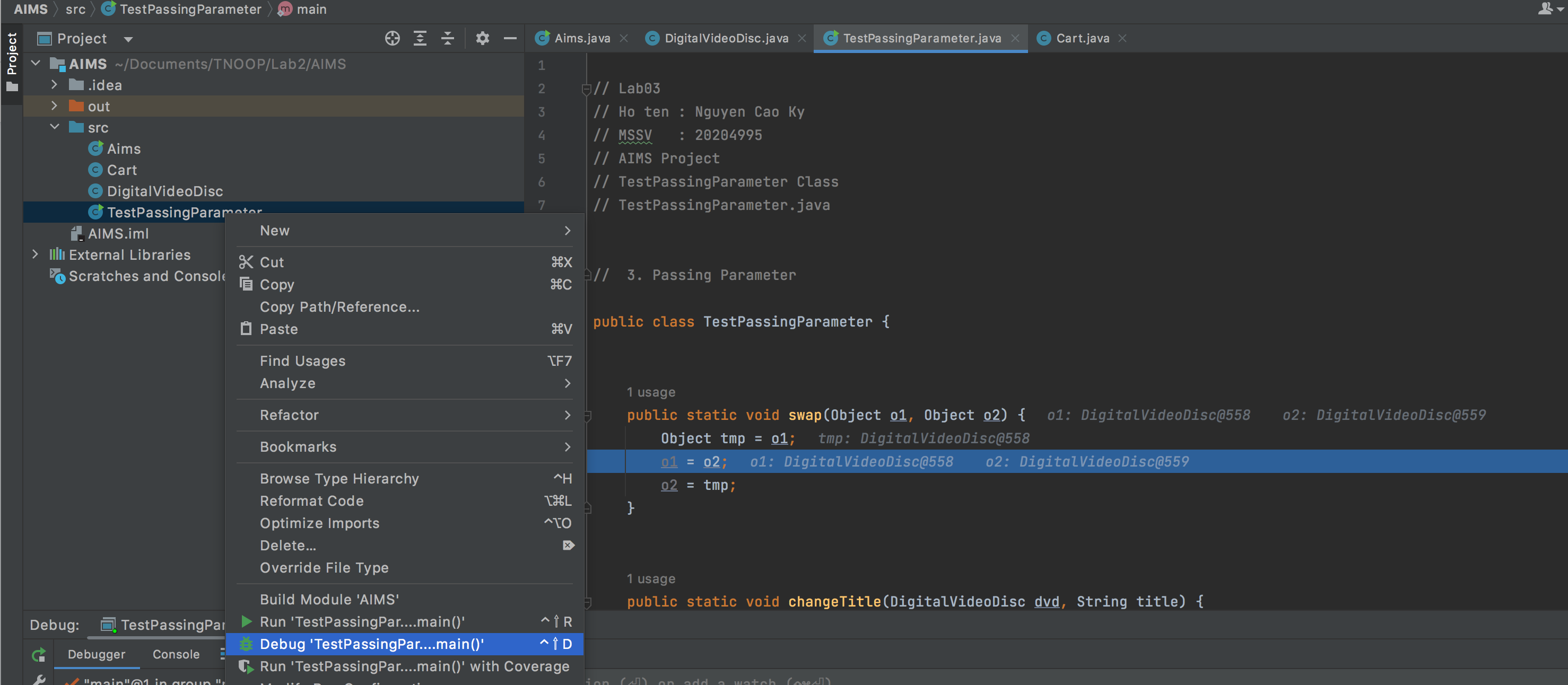


# 4. Use debug run

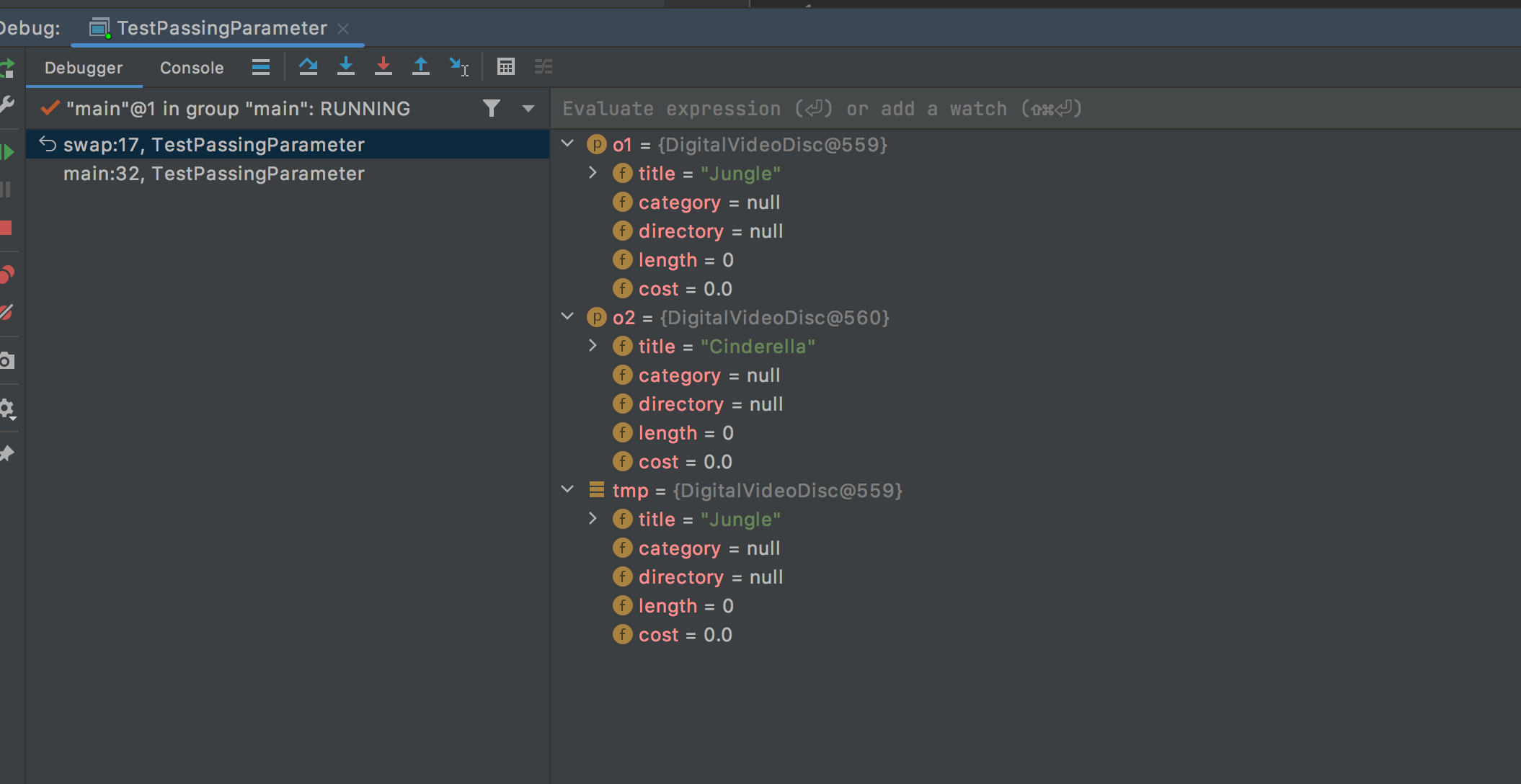
## 4.1 **Setting, deleting & deactivate breakpoints:**



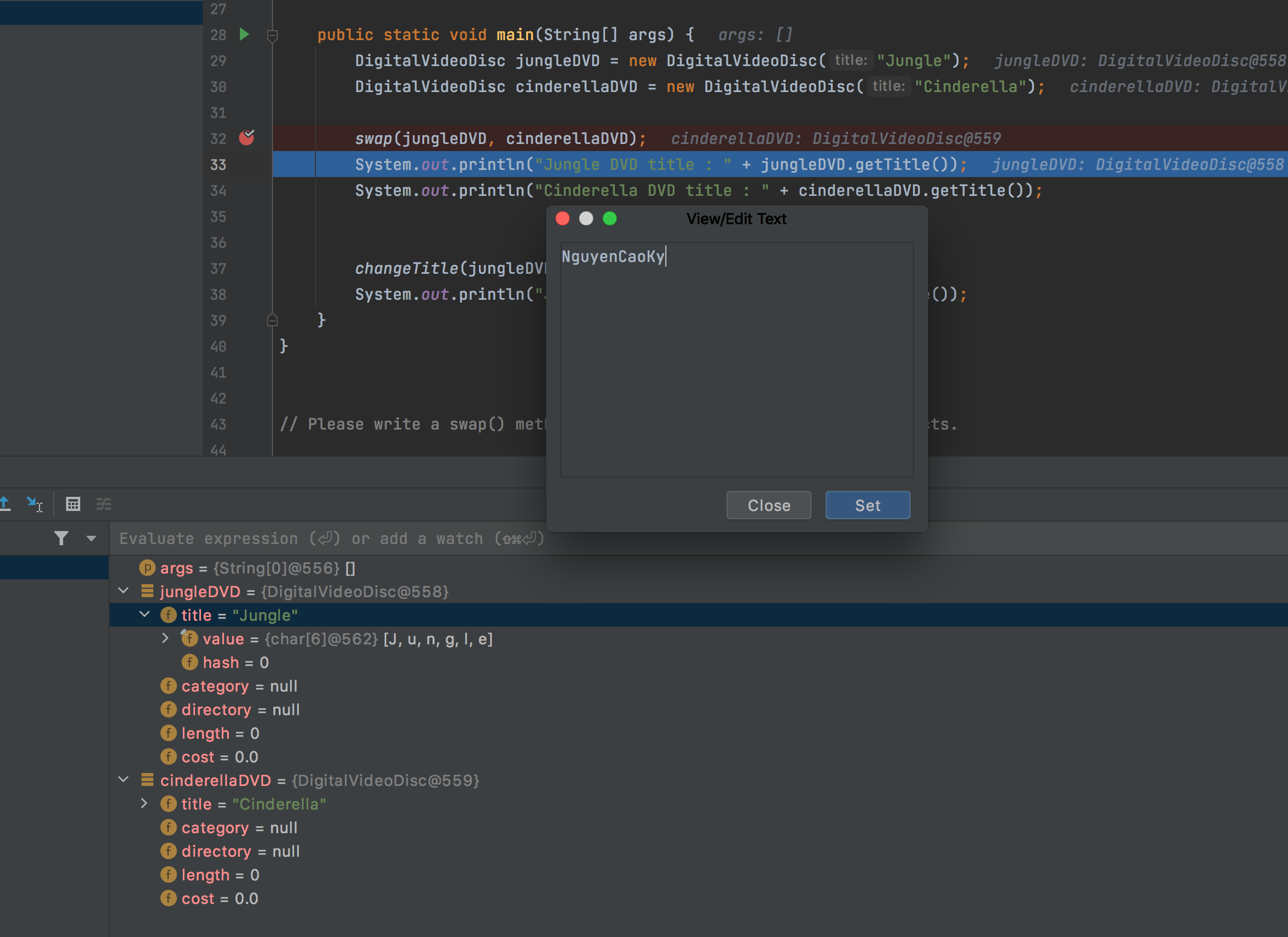
## 4.2 **Run in Debug mode**



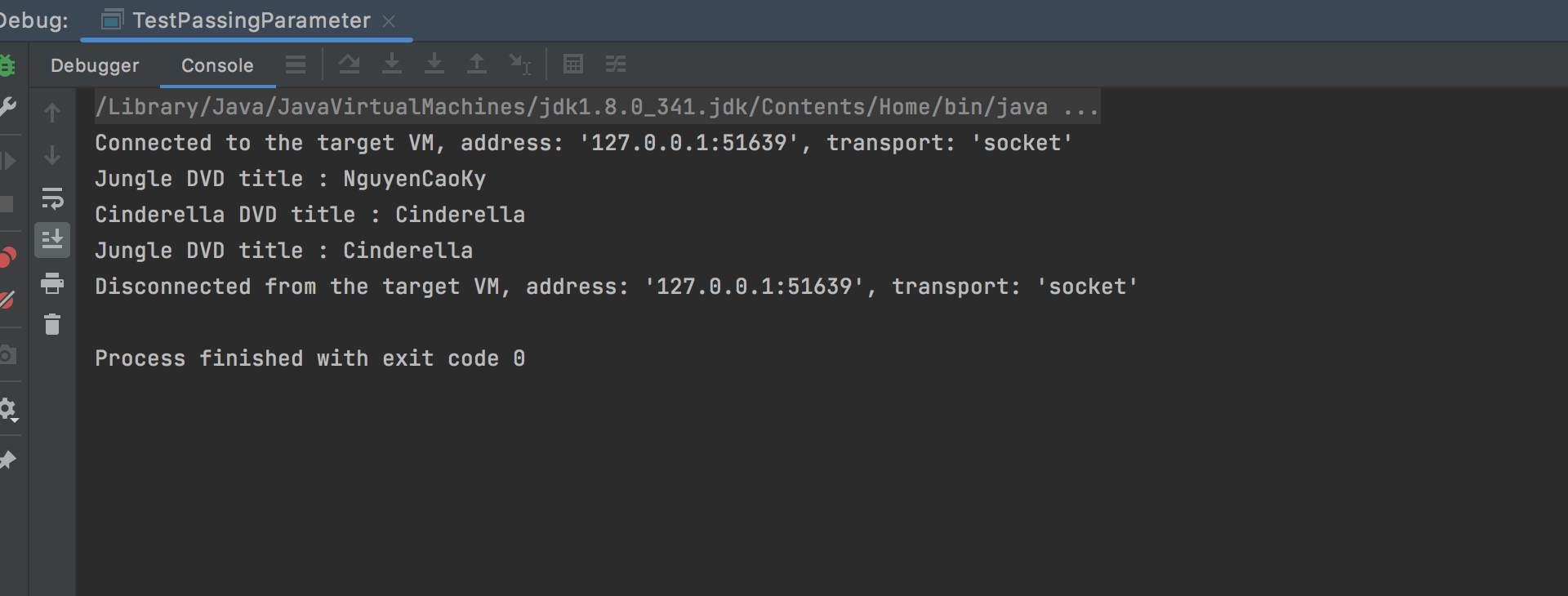
## 4.3 **Investigate value of variables:**



## 4.4 **Change value of variables**

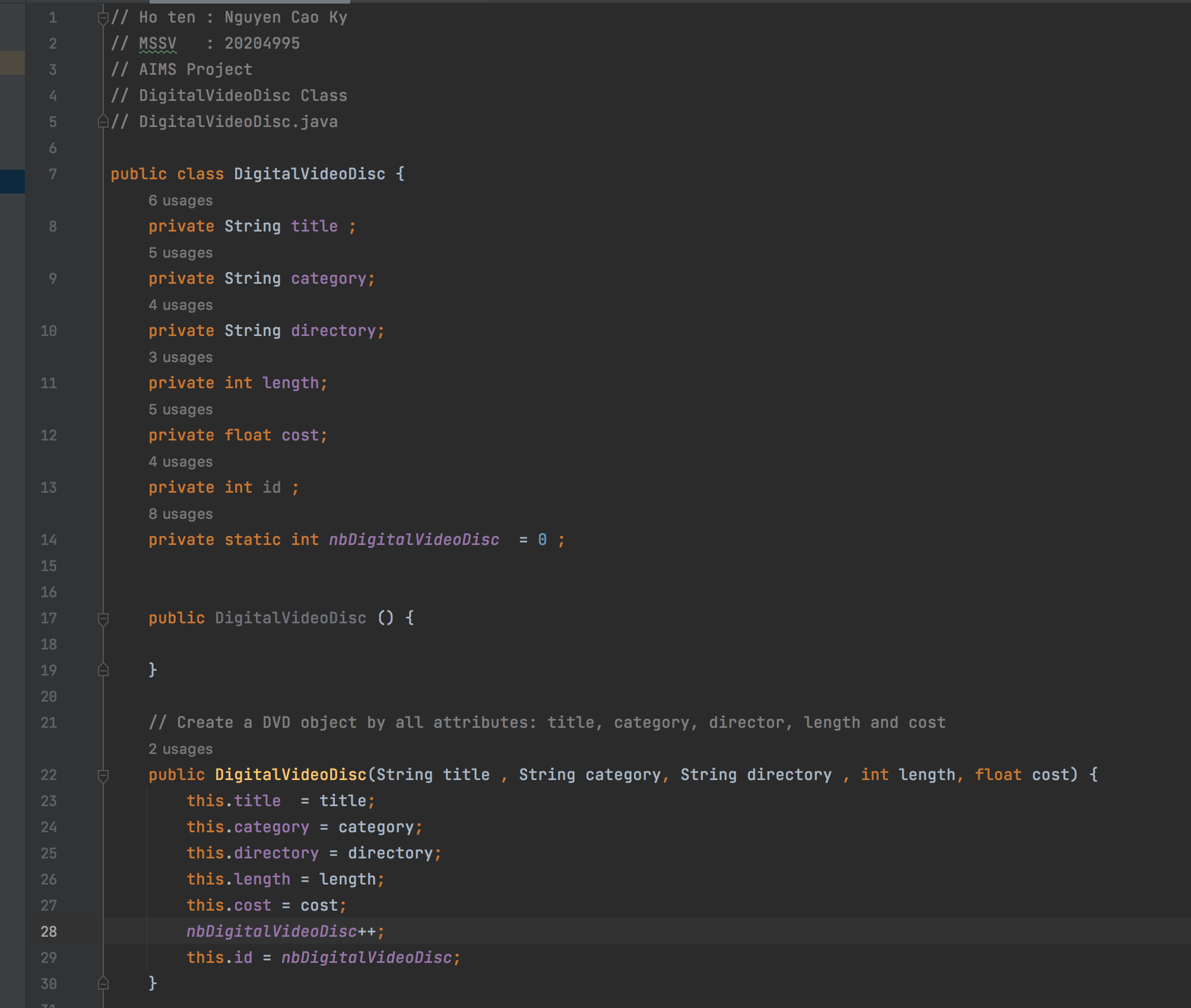


*Kết quả*

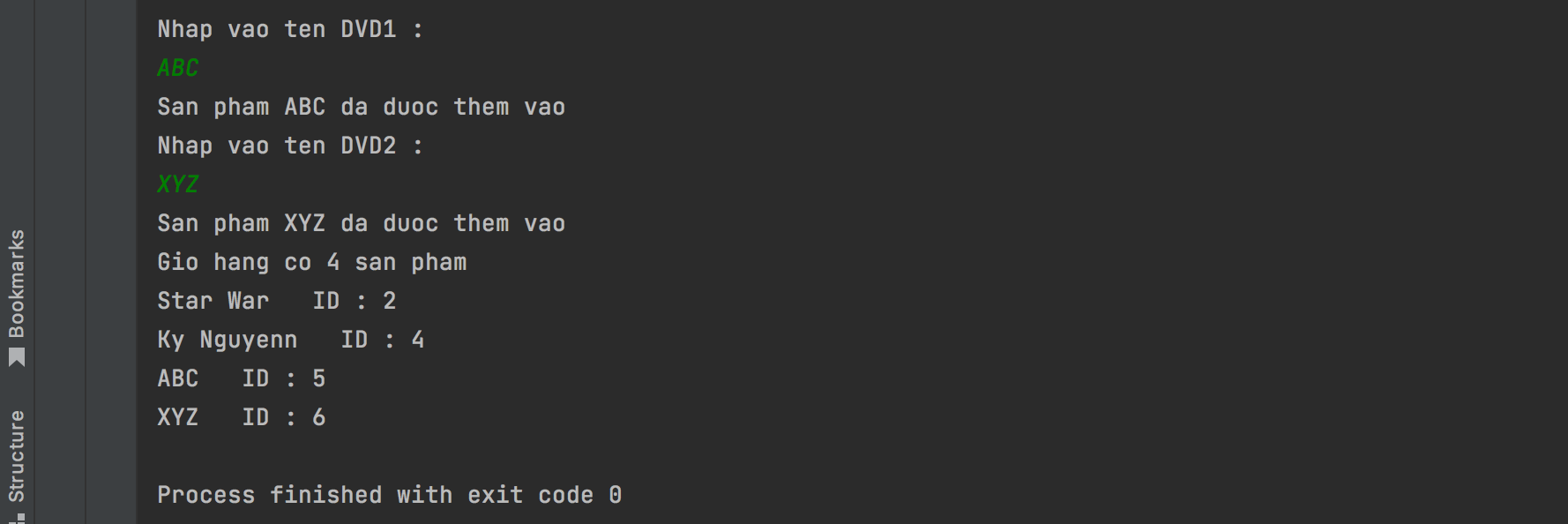
**

# 5. Classifier Member and Instance Member

Tạo một thuộc tính nbDigitalVideoDisc thuộc lớp sao cho mỗi khi một đối tượng mới được khởi tạo thì nbDigitalVideoDisc sẽ tăng 1 và được gán cho thuộc tính id của đối tượng

**

*Kết quả*

**

# 6. Create accessors and mutators for the class DigitalVideoDisc

# Open the **Cart** class

Write new methods to implement the following functions:

- Create a new method to print the list of ordered items of a cart, the price of each item, and the total price. Format the outline as below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CART\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ordered Items:

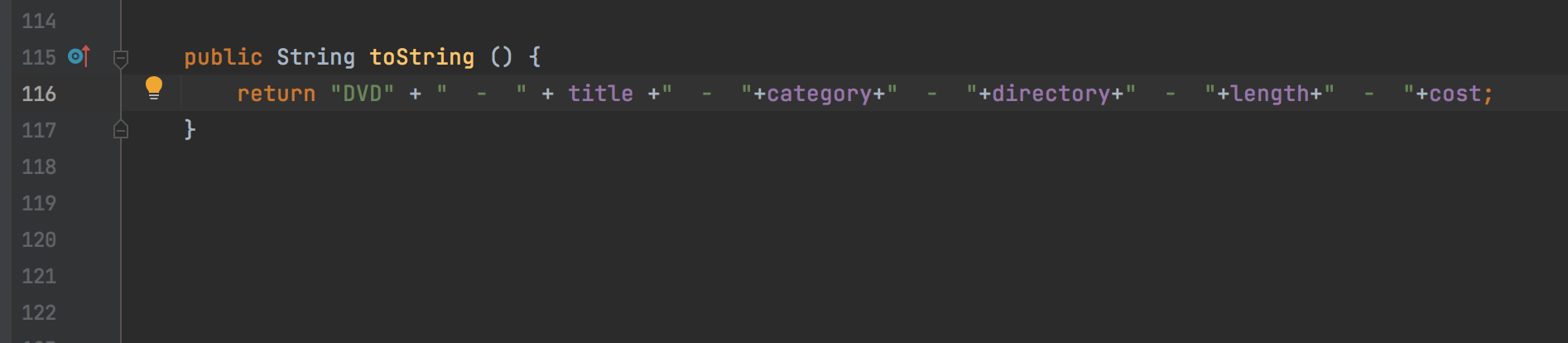
1. DVD - [Title] - [category] - [Director] - [Length]: [Price] $

2. DVD - [Title] - ...

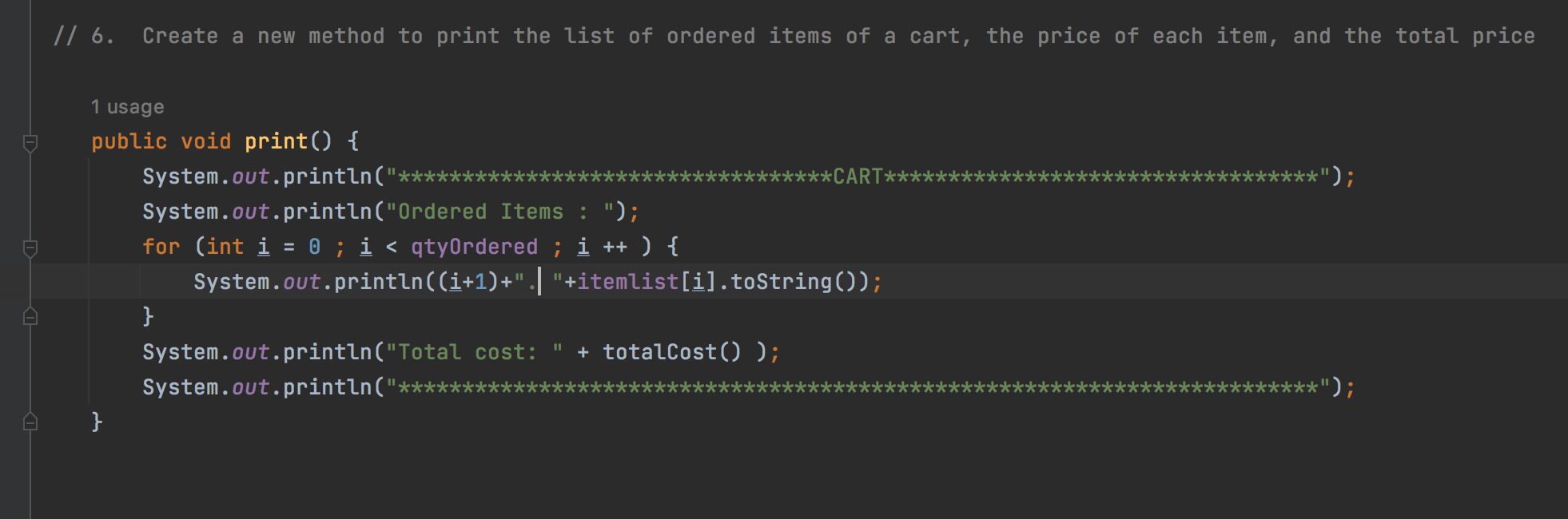
Total cost: [total cost]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*toString method:*

**

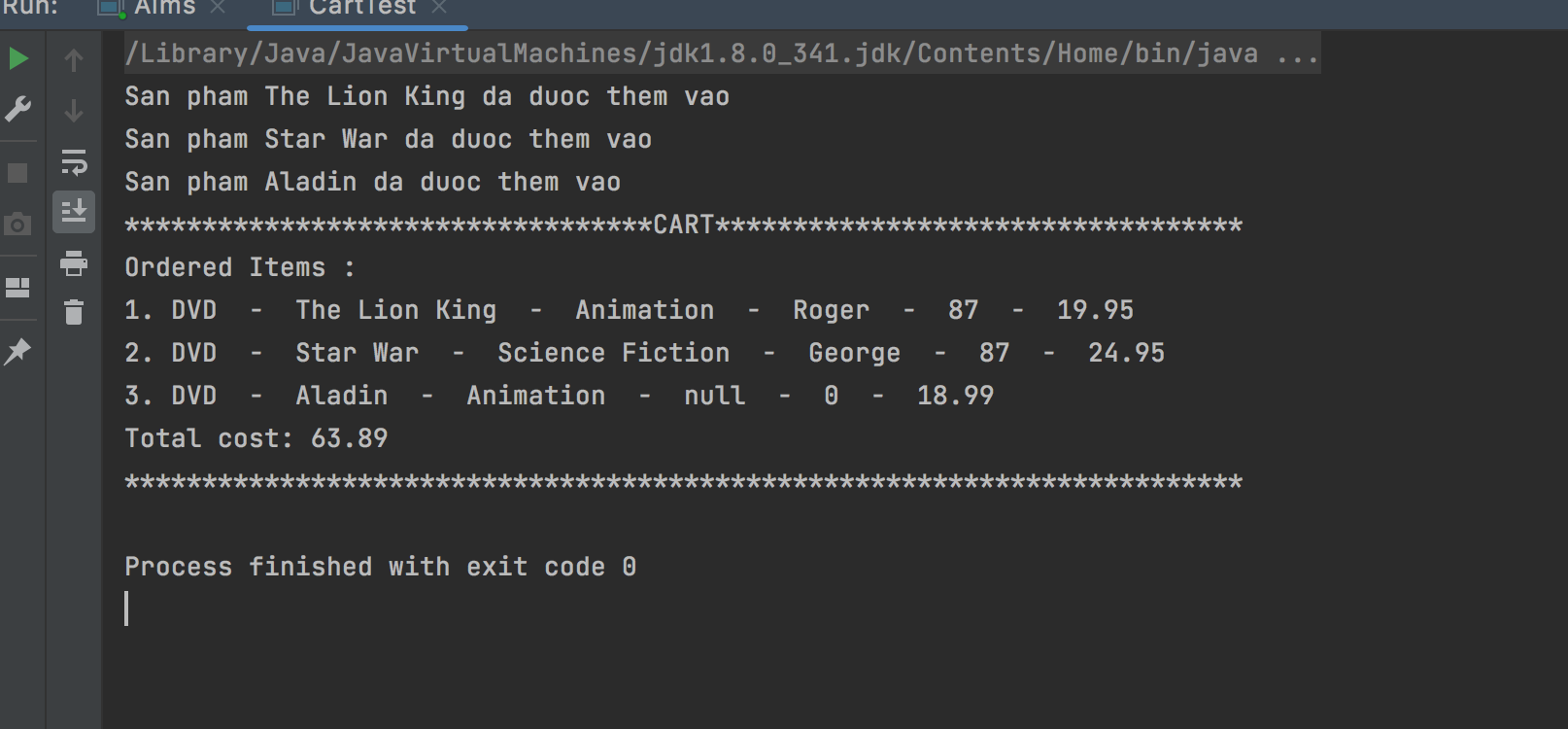
*print method:*

**

*CartTest class:*

**

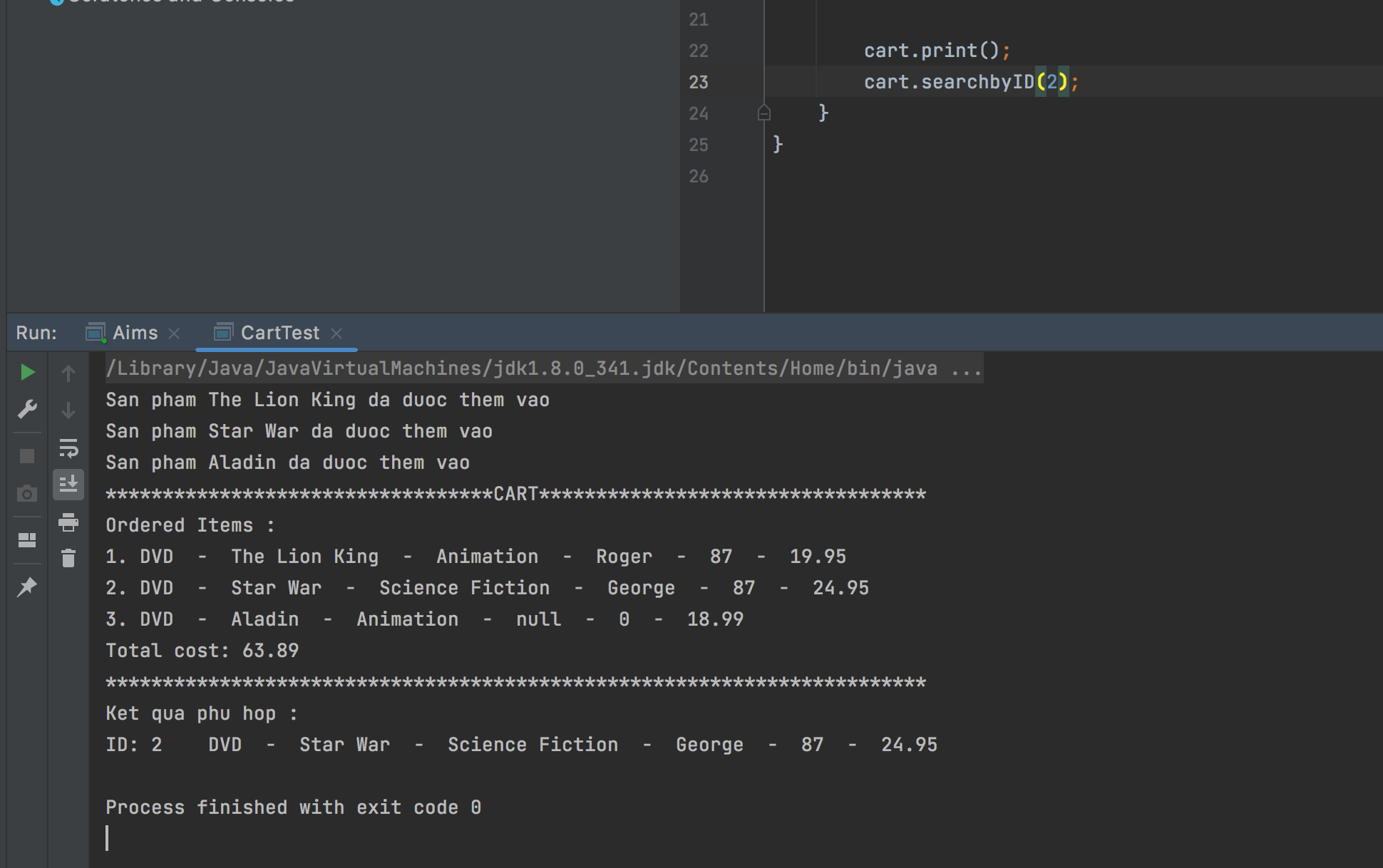
*Kết quả :*

**

- Search for DVDs in the cart by ID and display the search results. Make sure to notify the user if no match is found.

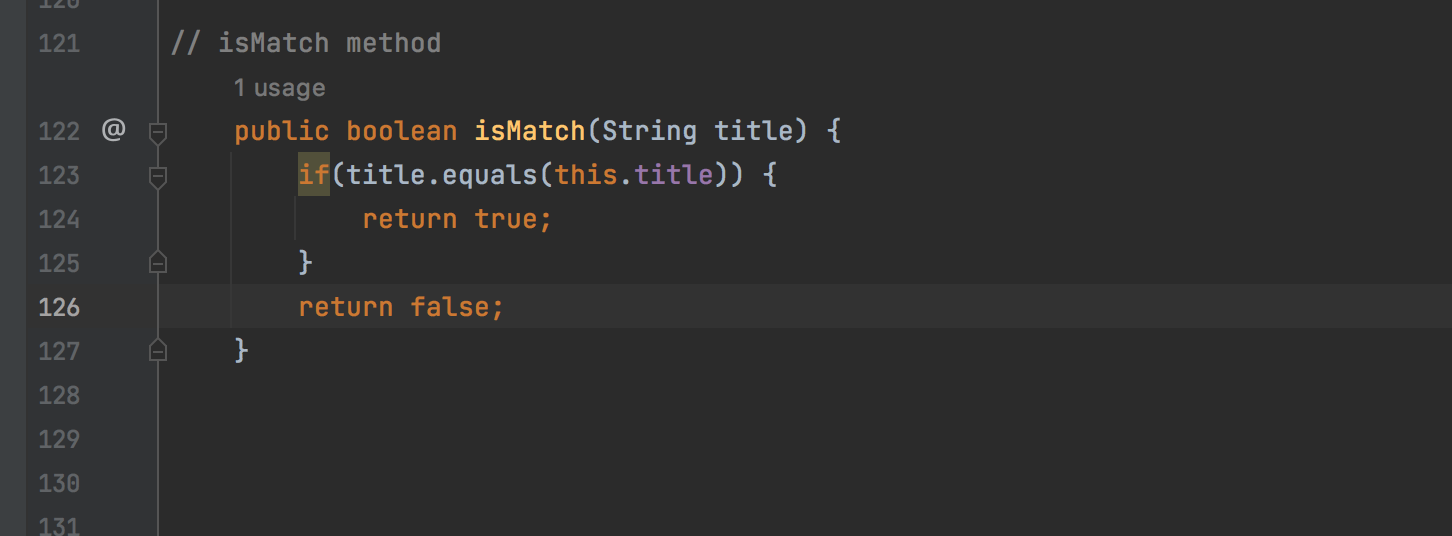
**

*Kết quả :*

**

- Search for DVDs in the cart by title and print the results. Make sure to notify the user if no match is found. ***Refer to problem statement in Lab02 for the matching rule.* Suggestion**: write a **boolean isMatch(String title)** method in the **DigitalVideoDisc** which finds out if the corresponding disk is a match given the title.

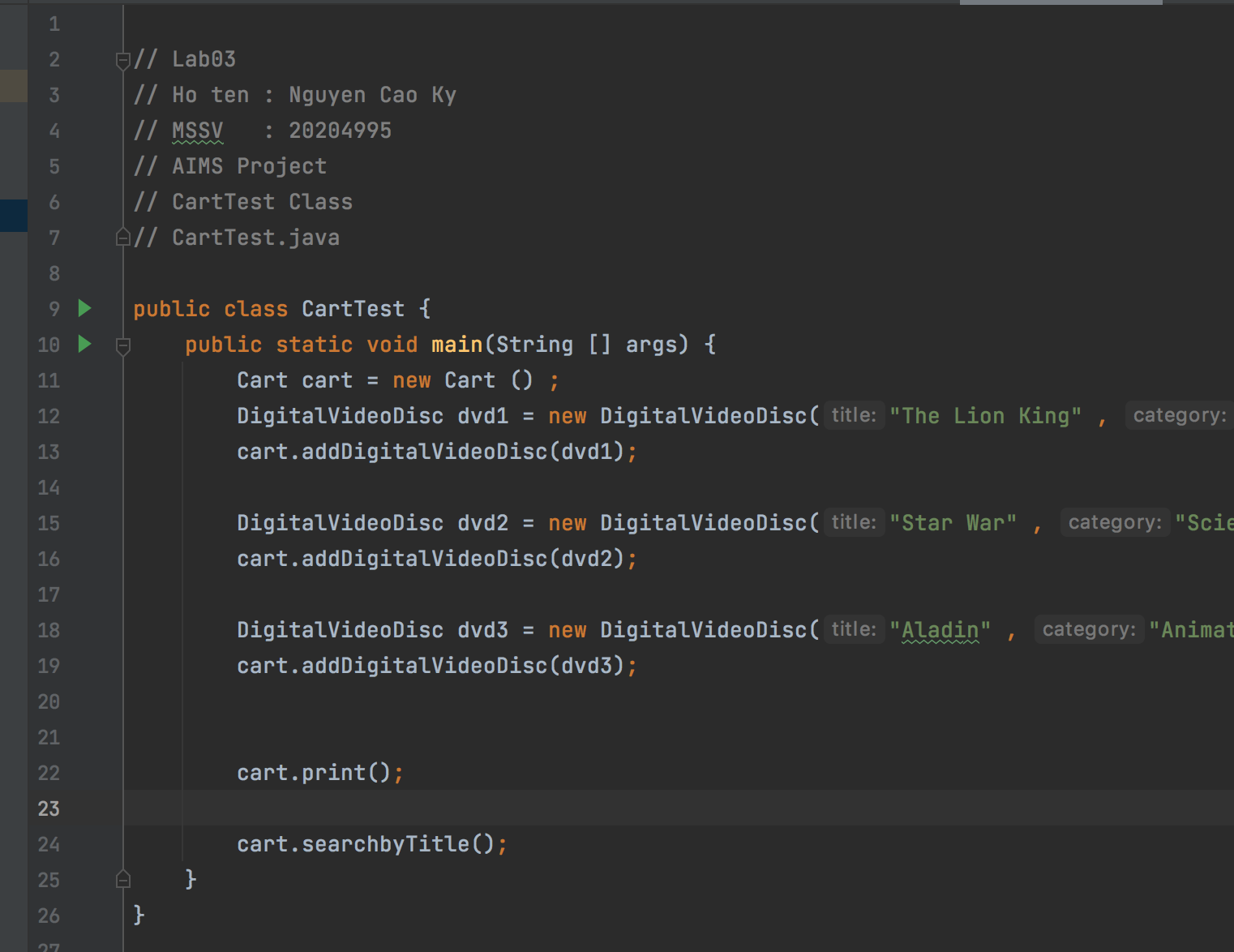
*isMatch method:*

**

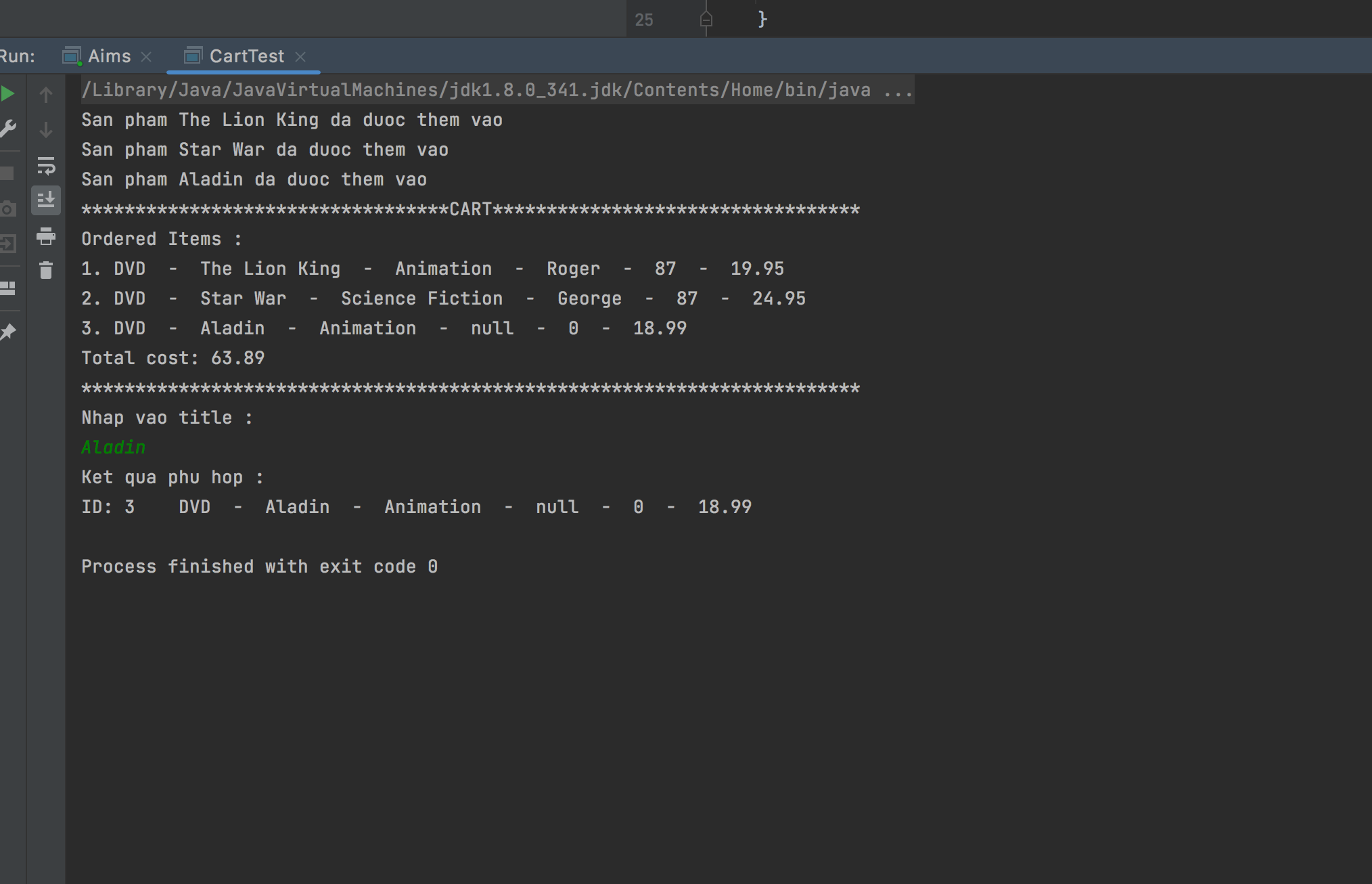
*searchbyTitle method:*

**

*CartTest:*

**

*Kết quả :*

**

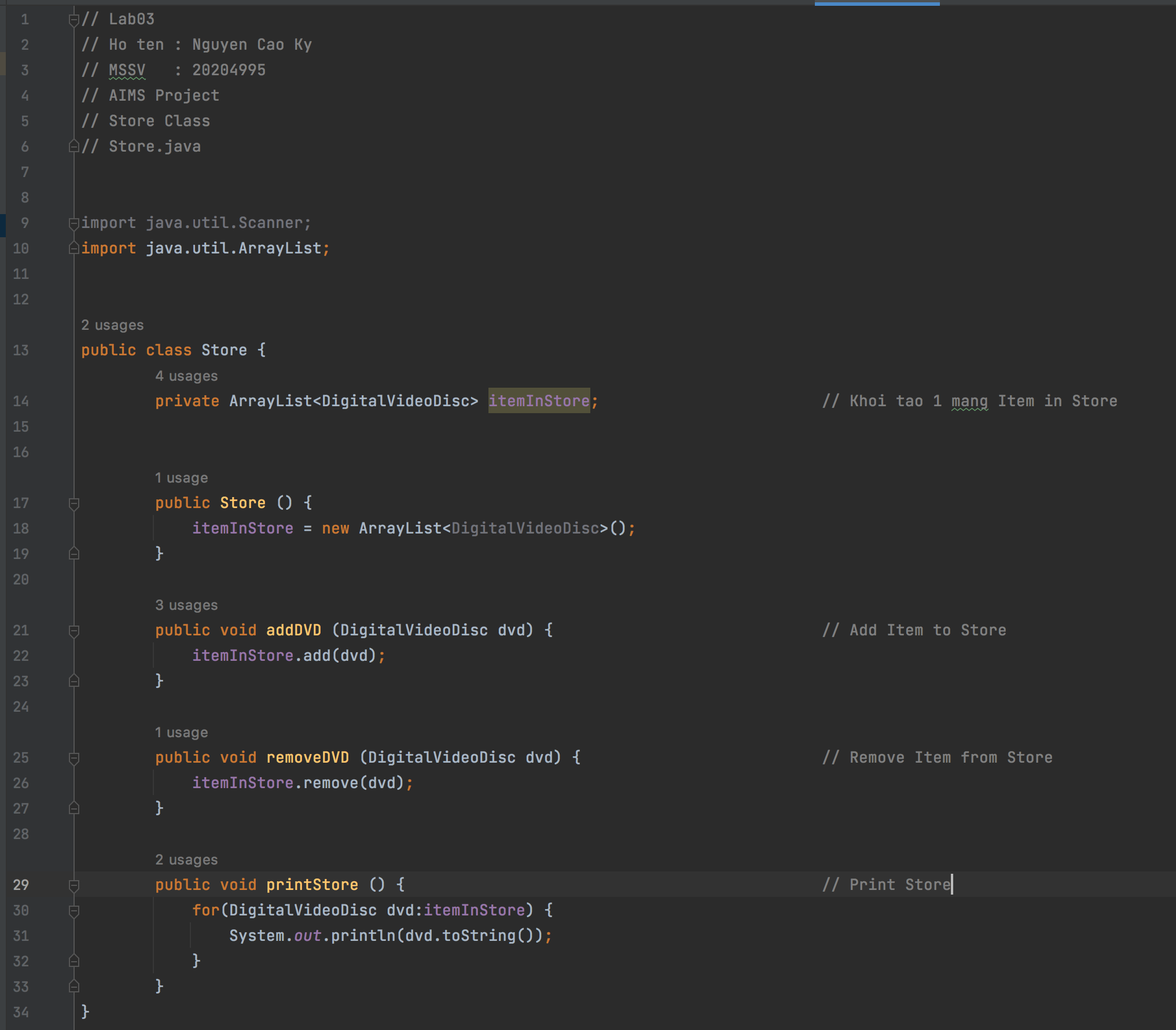
# 7. Implement the **Store** class

- Create a **Store** class, which contains one attribute **itemsInStore[]** – an array of DVDs available in the store.

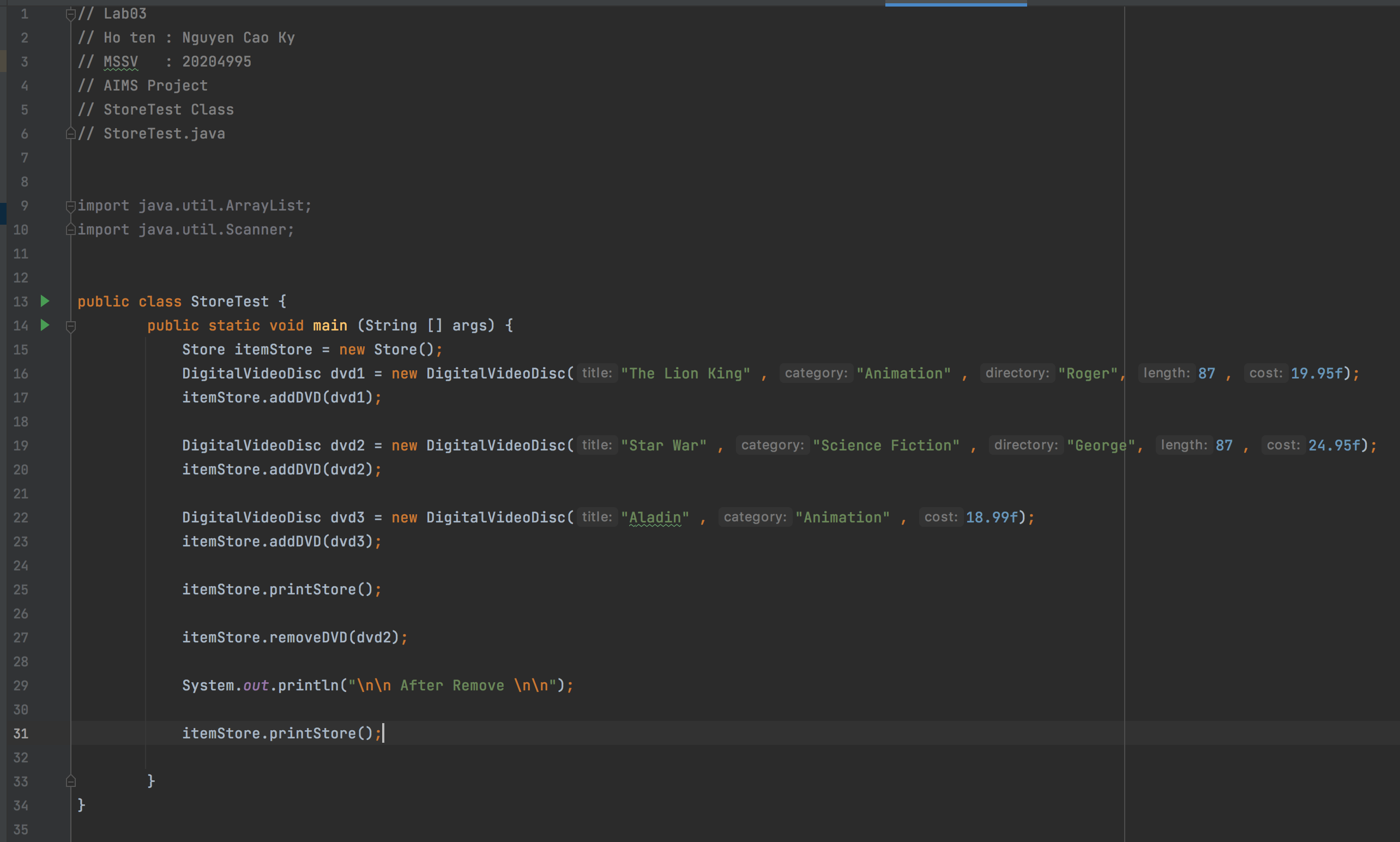
- To add and remove DVDs from the store, implement two methods called **addDVD** and **removeDVD**

- Test these two methods in **StoreTest** class.

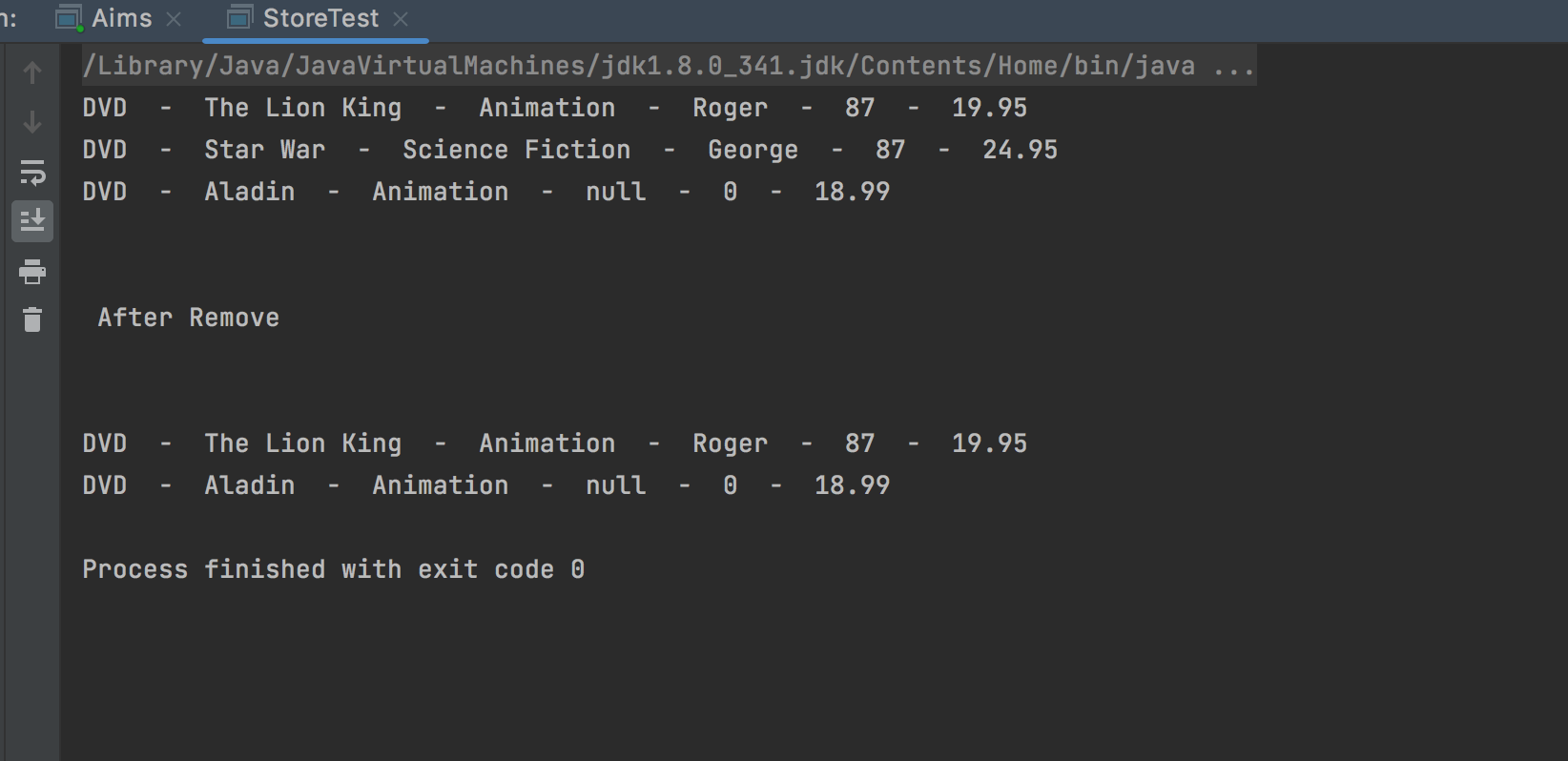
*Store Class:*

**

*StoreTest Class:*

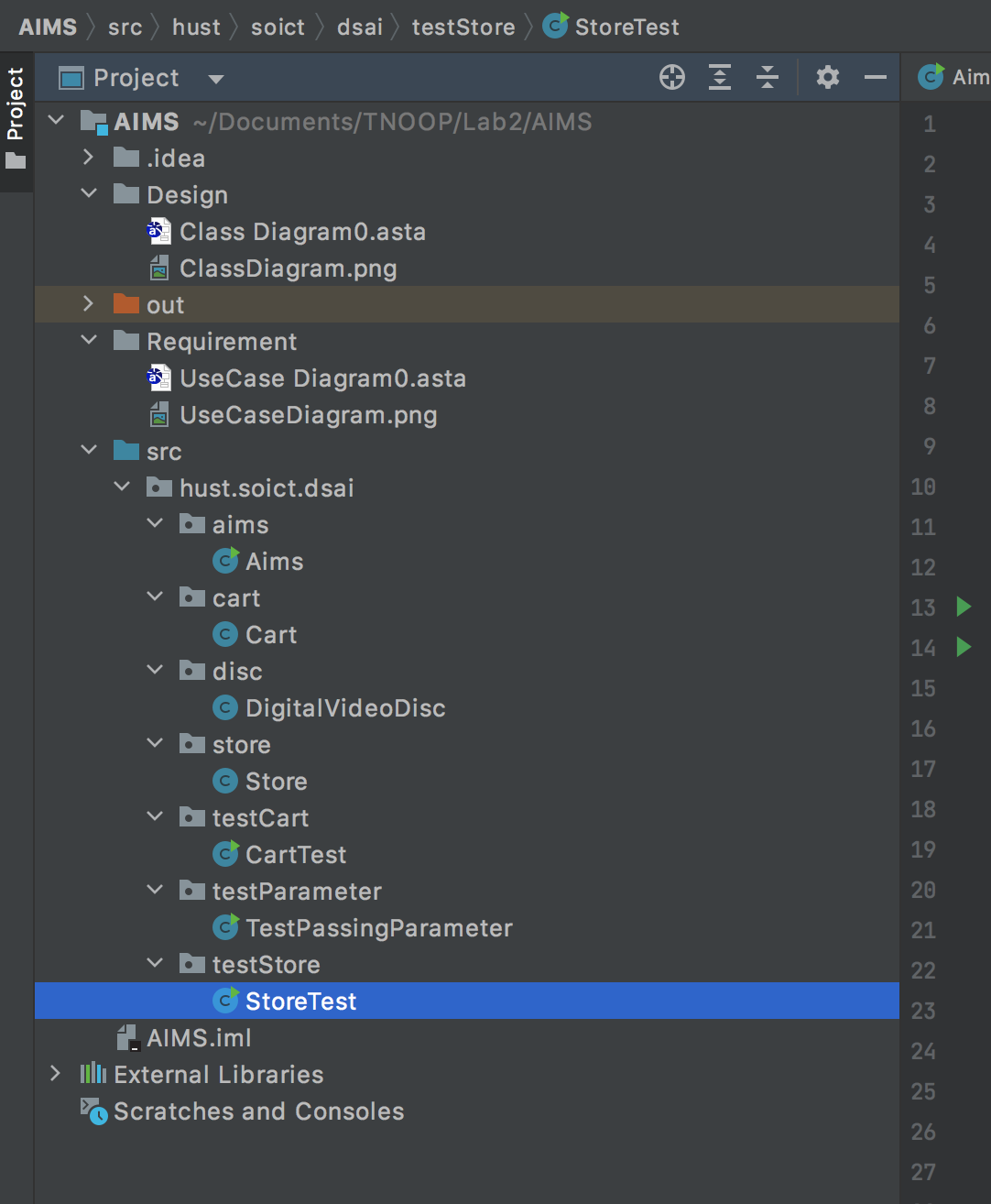
**

*Kết quả :*

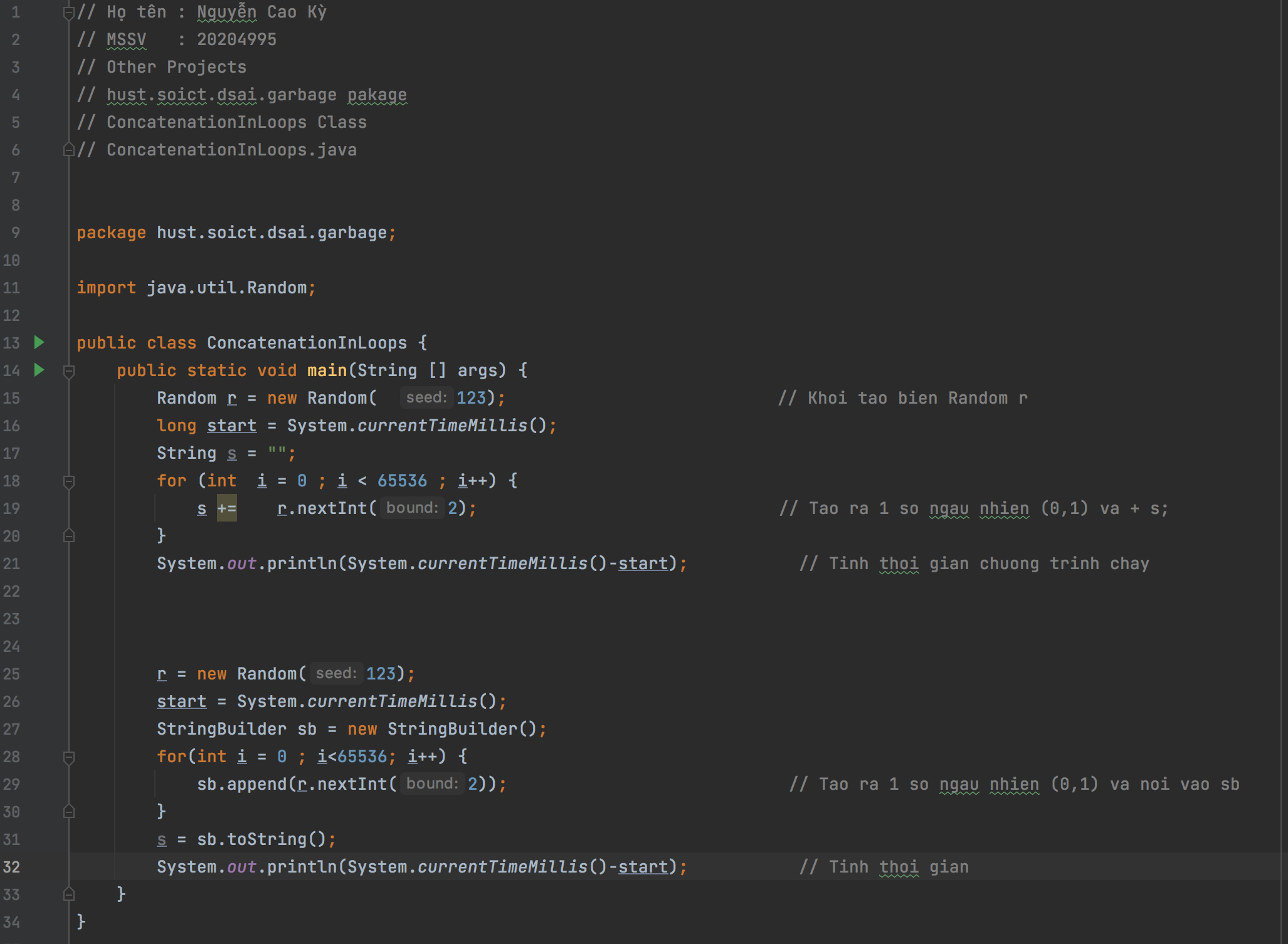
**

# 8. Re-organize your projects

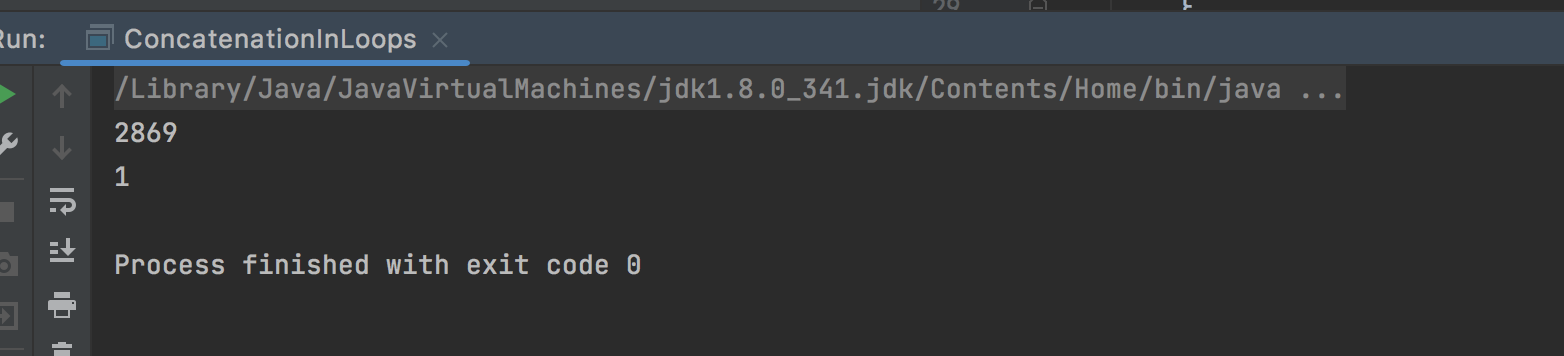
*Kết quả :*

**

# 9. **String**, **StringBuilder** and **StringBuffer**

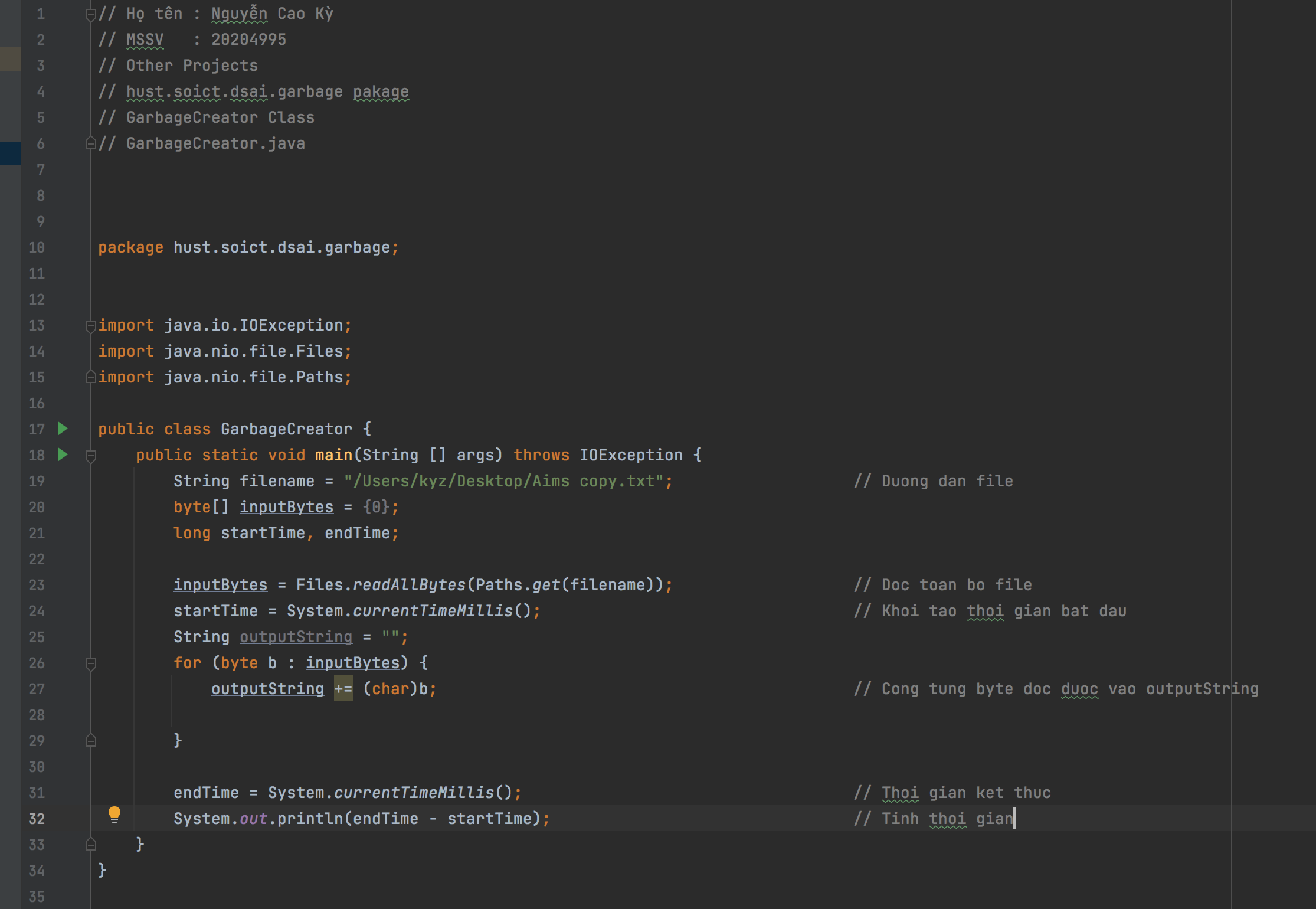


Kết quả:

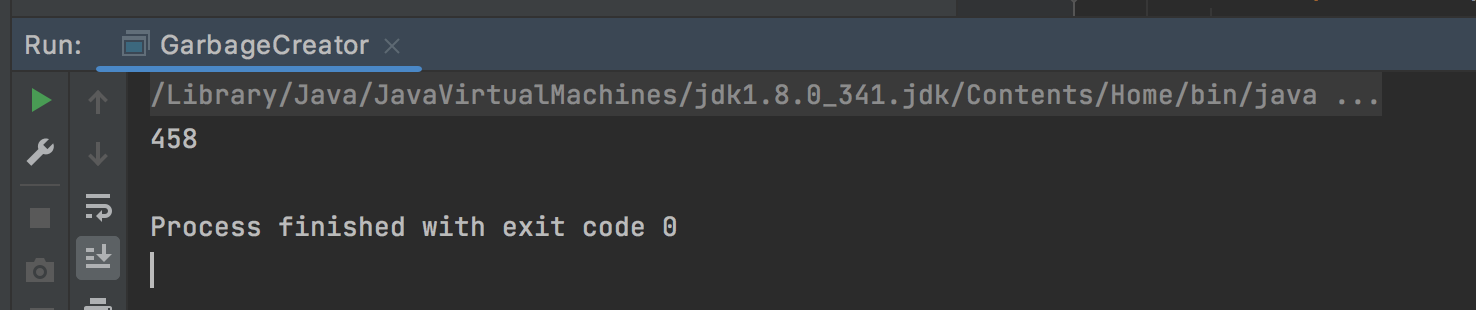


-> Sử dụng “+” mất rất nhiều thời gian và sử dụng lớp StringBuffer(StringBuilder) cải thiện rất nhiều.

*GarbageCreator Class:*

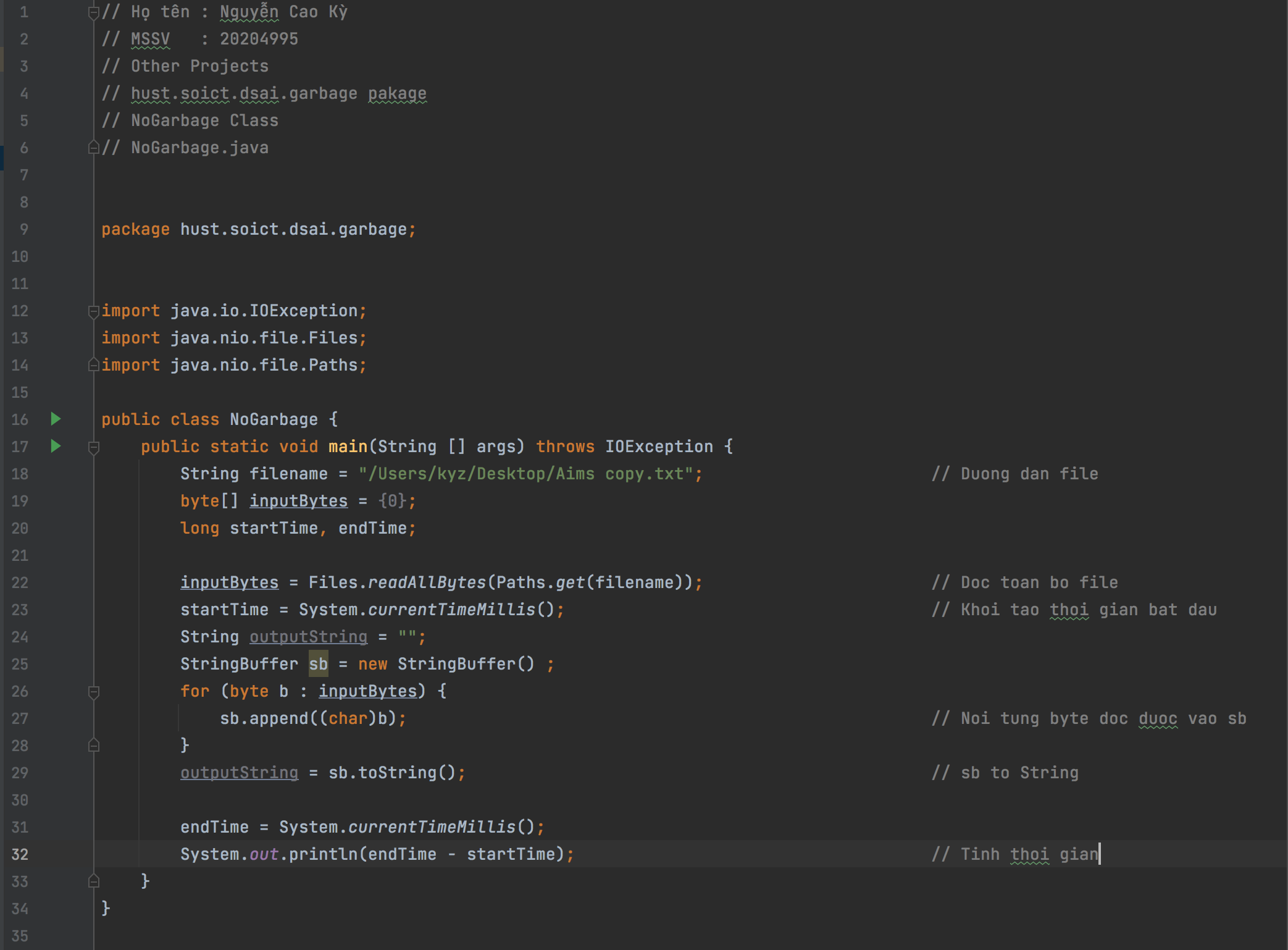
**

*Kết quả :*

**

*NoGarbage Class:*

Cải thiện thời gian thực hiện chương trình bằng StringBuffer



Kết quả:

